

时间敏感网络端处理逻辑 TSE 设计方案 (版本 1.2)

OpenTSN

OpenTSN 开源项目组

2021 年 6 月

版本历史

| 版本 | 修订时间 | 修订内容 | 修订人 | 文件标识 |
|-----|-----------|--|-----|------------|
| 1.0 | 2021.5 | 初版编制 | | OpenTSN3.0 |
| 1.1 | 2021.6 | 总体架构框图中：添加网络接收处理到网络发送处理逻辑的连线，为 PTP 报文路径 | | |
| 1.2 | 2021.6.24 | 1.总体架构框图 2-1 中：添加一个 GMII 网络接口，用于内部连接 HCP； 2.更新表 2-1：删除架构图中没有体现的信号 | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

目录

| | |
|-------------------------------------|----|
| 1、概述 | 4 |
| 2、总体设计 | 4 |
| 2.1. 总体架构 | 4 |
| 2.1.1 主机接收处理逻辑 | 5 |
| 2.1.2 主机发送处理逻辑 | 7 |
| 2.1.3 网络输入处理逻辑 | 7 |
| 2.1.4 网络输出处理逻辑 | 9 |
| 2.1.5 内部处理逻辑 | 10 |
| 2.2 帧的处理流程 | 11 |
| 2.2.1 主机口进网络口出的帧处理流程 | 11 |
| 2.2.2 网络口进主机口出的帧处理流程 | 12 |
| 2.2.3 网络口进网络口出的帧处理流程 | 14 |
| 附录 A：数据格式定义 | 15 |
| 附录 B：内部表项格式定义 | 18 |
| 附录 C：内部寄存器定义 | 19 |
| 附录 D：NMAC 报文格式 | 25 |
| 附录 E：command/command_ack 命令格式 | 28 |

1、概述

OpenTSN2.0 开源硬件逻辑既可作为 TSN 网卡又可作为 TSN 交换机使用，为了简化其作为 TSN 交换机使用时的逻辑复杂度以及增强其作为 TSN 网卡使用时的功能可扩展性，并且考虑到逻辑模块的复用，现将 OpenTSN2.0 开源硬件逻辑拆分为时间敏感网络硬件控制逻辑（HCP）、时间敏感网络端处理逻辑(TSE)、时间敏感网络交换处理逻辑(TSS)三个模块。本文详细介绍时间敏感网络端处理逻辑 TSE 的设计方案，TSE 主要负责将标准以太网报文映射为 TSN 报文并将 TSN 报文逆映射为标准以太网报文。

2、总体设计

2.1. 总体架构

时间敏感网络端处理逻辑的总体架构框图如图 2-1。

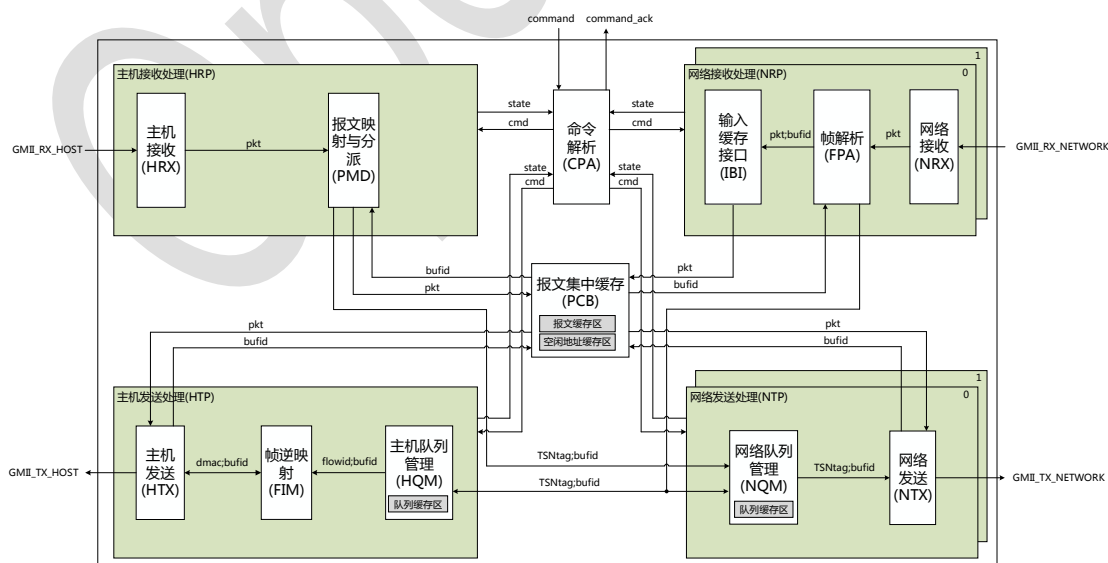


图 2-1 总体架构框图

总体架构框图中的信号含义如下表 2-1。

表 2-1 TSE 总体架构框图中信号含义

| 信号 | 位宽 | 含义 |
|-------------------------|-----|---------------------------|
| pkt_134（报文集中缓存接收或发送的信号） | 134 | 报文体数据，具体格式参考附录 A |
| pkt_9（非报文集中缓存模块的信号） | 9 | 报文体数据，具体格式参考附录 A |
| dmac | 48 | 报文目的 mac 地址 |
| pkt_bufid | 9 | 报文数据在报文缓存区中缓存的 ID 号 |
| flowid | 14 | 流 id，报文在 tsn 网络中的流标识 |
| TSNtag | 48 | 携带报文类型、流 id、注入地址、提交地址的描述符 |
| cmd | xx | 配置命令，包括寄存器及表项的配置 |
| state | xx | 各个模块的状态上报信息 |
| command | 204 | 读、写命令 |
| command_ack | 204 | 读命令响应 |

将整个架构划分为五大部分逻辑：主机接收处理逻辑、主机发送处理逻辑、网络输入处理逻辑、网络输出处理逻辑、以及内部处理逻辑。下面对该五部分逻辑进行逐一介绍。

2.1.1 主机接收处理逻辑

主机接收处理逻辑的内部组成框图如图 2-2。

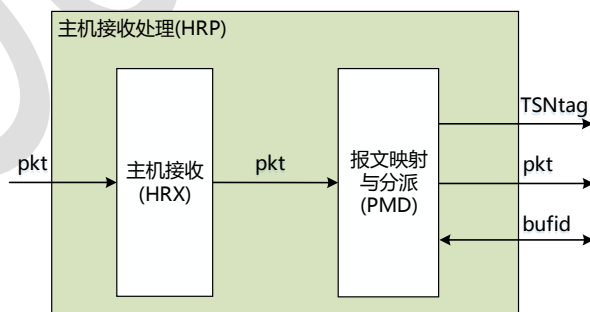


图 2-2 HRP 模块组成框图

HRX(Host RX)网络接收模块：主要功能是接收主机接口传来的报文，将报文传输时钟域由 GMII 接收时钟域切换为架构内部时钟域，

在端处理逻辑初始化时，丢弃所有的报文；在初始化完成后，才传输所有的报文。

PMD(Packet Mapping and Dispatching, 报文映射与分派)模块:

主要功能是实现报文位宽转换，报文监管，报文缓存，查表映射。报文位宽转换是将报文每拍数据位宽由 9bit 转换为 134bit；报文监管是根据 bufid 数量与未映射报文阈值大小关系来决定是否对未经映射报文进行传输，即当 bufid 剩余数量少于未映射报文阈值时，将未映射报文进行丢弃；报文缓存是指将报文集中缓存模块分配的 bufid 转换为报文缓存基地址，将报文写入到集中缓存区中；查表映射是指 TCP/UDP 报文用五元组去查找映射表，将 bufid 和查表结果 TSntag 传输给网络接口；IP 报文但非 TCP/UDP 报文不查映射表，直接将 bufid 和 DMAC 字段值（即 TSntag 语义）传输给网络接口；非 IP 报文不查转发表，直接传输给硬件控制点 HCP。

表 2-2 TSntag 格式定义

| 位宽 | 名称 | 位置 | 描述 |
|----|--------------|---------|--|
| 3 | Flow type | [47:45] | 流类型。其中 3'h0 表示 ST 分组，3'h1 表示 ST 分组，3'h2 表示 ST 分组，3'h3 表示 RC 分组，3'h4 表示 PTP 分组，3'h5 表示 NMAC 分组，3'h6 表示 BE 分组，3'h7 表示需重组分组 |
| 14 | Flow id/IMAC | [44:31] | 静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换架构命中同一条表项 |
| 16 | Seq id | [30:15] | 用于标识每条流中报文的序列号 |
| 1 | Frag flag | [14] | 用于标识分片后的尾。1'h0 表示第一个分片或中间分片，1'h1 表示最后一个分片 |
| 4 | Frag ID | [13:10] | 用于表示当前分片报文在原报文中的分片序列号 |
| 5 | inject addr | [9:5] | ST 流在源端等待发送调度时缓存地址 |
| 5 | submit addr | [4:0] | ST 流在终端等待接收调度时缓存地址 |

2.1.2 主机发送处理逻辑

主机发送处理逻辑的内部组成框图如图 2-3。

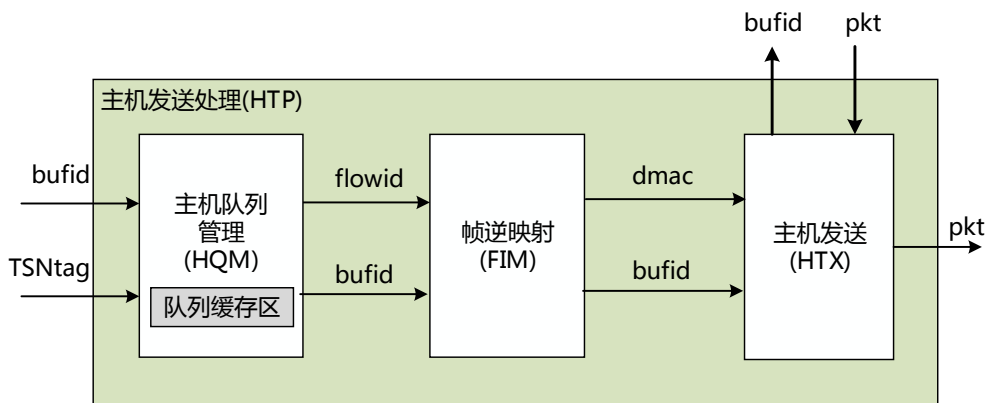


图 2-3 HTP 模块组成框图

HQM(Host Queue Manage,主机队列管理)模块： 主要功能是对从主机接口输出的报文描述符进行缓存管理，等待调度输出。

FIM(Frame Inverse Mapping 帧逆映射) 模块： 主要功能是将接收到的报文查找逆映射表，获得报文真实的目的 MAC。

HTX(Host TX，主机发送)模块： 主要功能是从报文集中缓存区中读取报文并释放 pkt_bufid、将报文从主机接口输出。从集中缓存区中读取报文时，将 pkt_bufid 映射成报文在缓存区中缓存的基地址，并根据此地址从报文集中缓存区中读出报文，在报文最后一拍数据从缓存区中读出时，将 pkt_bufid 进行释放以便后续进入架构的报文使用，在将报文从 GMII 接口输出时，需把报文的位宽由 134bit 转换为 8bit 并增加帧前导码、帧开始符。

2.1.3 网络输入处理逻辑

网络输入处理逻辑的内部组成框图如图 2-4。

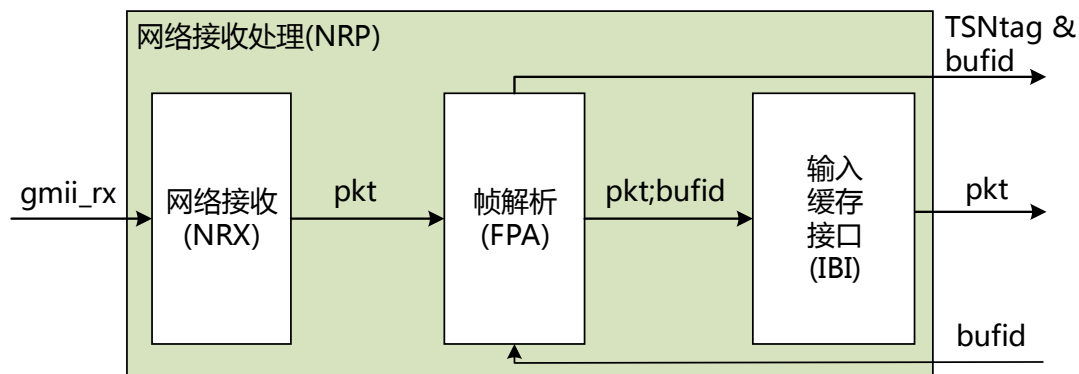


图 2-4 NRP 模块组成框图

NRX(Network RX, 网络接收)模块：主要功能是接收网络接口传来的报文，将报文传输时钟域由 GMII 接收时钟域切换到架构内部时钟域，以及记录架构接收时间同步报文的时间戳，并存放在 TSNTag 中。根据端处理逻辑所处的阶段来决定是否传输报文：若端处理逻辑处于初始化阶段，丢弃所有报文；若端处理逻辑处于配置阶段，只传输配置报文；端处理逻辑处于时钟同步阶段，传输非时间敏感报文；端处理逻辑处于正常工作阶段，接收所有报文。

FPA(Frame Parse, 帧解析)模块：主要功能是将报文位宽由 9bit 转换为 134bit；从帧的 TSNTag 中提取信息用来构造报文描述符；对 RC 报文和 BE 报文进行监管，若 bufid 剩余数量少于 RC 报文阈值时，将 RC 报文和 BE 报文均丢弃，若 bufid 剩余数量少于 BE 报文阈值时，将 BE 报文丢弃；并根据报文以太网类型对报文描述符进行分派，即将 ARP 报文描述符或映射后的 IP 报文描述符传输给主机接口，将来自 HCP 的映射后 PTP 报文描述符或 TSMP 报文描述符传输给网络接口，将来自交换机的映射后 PTP 报文描述符或 TSMP 报文描述符传输给网络接口 HCP。

IBI(Input Buffer Interface, 输入缓存接口)模块：主要功能是将

报文发送给报文集中缓存模块进行缓存。本模块接收帧解析模块的 134bit 数据使用两个寄存器进行缓存，其中任何一个寄存器有数据则往报文集中缓存模块发出写请求，同时将接收的数据写入另一个寄存器中。

2.1.4 网络输出处理逻辑

网络输出处理逻辑的内部组成框图如图 2-5。

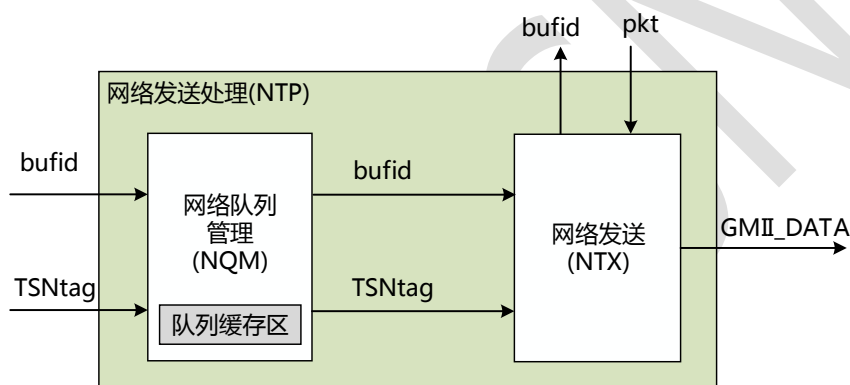


图 2-5 NTP 模块组成框图

NQM(Network Queue Management, 网络队列管理)模块：主要功能是将报文描述符写入到网络队列管理模块中进行缓存，在网络接口空闲时，将队列中的描述符调度输出给网络发送模块。

NTX(Network TX, 网络发送)模块：主要功能是从报文缓存区中读取报文并释放 pkt_bufid、计算时间同步报文的透明时钟并更新透明时钟域，将报文传输时钟域由结构内部时钟域切换到 GMII 发送时钟域。从缓存区中读取报文时，将 pkt_bufid 映射成从缓存区读取报文的基地址，然后根据此地址往报文集中缓存模块读取报文，在从缓存区中读出最后一拍数据后将 pkt_bufid 进行释放以便后续进入架构的报文使用，在将报文从 GMII 接口输出时，需把报文的位宽由 134bit

转换为 8bit 并添加帧前导符和帧开始符。

2.1.5 内部处理逻辑

PCB(Pkt Centralize Bufm_memory, 报文集中缓存)模块：主要功能是对每个接口分配 bufid，并基于时分复用技术，将每个接口接收的报文写入集中缓存区中进行缓存；并且从每个接口回收 bufid，并基于时分复用技术，将报文从缓存区中读出给各接口；本模块需对所有空闲的 pkt_bufid 进行缓存(空闲地址缓存区的格式如下表 2-4)；并且为每个分配的 pkt_bufid 设计一个寄存器，用来记录该 pkt_bufid 对应报文的输出接口个数，bufid 每释放一次，将该寄存器减一，当得到的结果为 0 时，才将该 pkt_bufid 释放，当得到的结果不为 0 意味着该 pkt_bufid 对应的报文为组播且该报文还未从所有需要输出的接口输出。报文缓存区容量为 1024KB，被划分为 512 个缓存块，每个缓存块容量为 2KB。

表 2-3 报文缓存区数据格式

| 地址[8:0] | 内容[133:0] |
|-------------|--------------|
| 0-127 | 第 1 个报文缓存块 |
| 128-255 | 第 2 个报文缓存块 |
| ... | .. |
| 65408-65535 | 第 512 个报文缓存块 |

表 2-4 空闲地址缓存区数据格式

| 名称 | 含义 | 备注 |
|----------------|--------------------------|----------------------|
| pkt_bufid[8:0] | 当前“报文缓存区”中空闲的报文缓存块 ID 号。 | 使用 RAM 进行实现，深度为 512。 |

CPA(Command Parse, 命令解析)模块：负责将对来自 HCP 的 command 命令（命令数据格式见表 2-5）进行解析，实现对本地寄存

器、映射表、逆映射表的配置；并且解析读命令，将读数据封装在读响应命令中输出给 HCP。

表 2-5 command/command_ack 命令格式

| 位置 | 位宽 | 名称 | 说明 |
|-----------|-----|----------------|---|
| [203:180] | 8 | node_id | 该字段用来标识对哪个节点进行读写。每个 TSE 或 TSS 都有一个唯一的节点 ID。该字段在 TSN 网卡+TSN 交换机模式下使用到。 |
| [179:172] | 8 | dest_module_id | 该字段用来标识对一个节点内的哪个模块进行控制。TSE 或 TSS 内部每个子模块都有一个唯一的模块 ID |
| [171:168] | 4 | type | 4'b0001:寄存器或表项的写命令； 4'b0010:寄存器或表项的读命令； 4'b0110:寄存器或表项的读响应。 |
| [167:152] | 16 | addr | 寄存器或表项的读/写地址 |
| [151:0] | 152 | data | 寄存器或表项的读/写数据；其中五元组映射表的表项位宽最大，为 152bit |

2.2 帧的处理流程

架构中包含 1 个主机口和 1 个网络口，主机接口与外部设备（需接入 TSN 网络中）连接，网络接口与 HCP 或 TSS 连接。下面详细介绍帧的处理流程。

2.2.1 主机口进网络口出的帧处理流程

主机口输入的帧先在主机接收模块（HRX）进行跨时钟域处理，并在每拍数据中增加 1bit 的头尾标识位，然后根据 TSE 所处的阶段决定是否传输报文，即 TSE 在处于初始化时，丢弃所有的报文；TSE 在初始化完成后，才传输所有的报文给帧映射与分派模块（FMD）。

帧映射与分派模块（FMD）接收到报文后，将报文集中缓存模块（PCB）分配的 bufid 转化为报文缓存的基地址，将报文写到报文

集中缓存区中；同时对报文进行映射：若报文为 TCP/UDP 报文（以太网类型为 0x0800，且 IP 协议为 0x6/0x11），则提取报文五元组，用五元组去查找映射表，获得查表结果；若报文非 TCP/UDP 报文，则直接将其 DMAC 字段作为查表结果；查表后将查表结果 TSntag 和 bufid 传给网络队列管理模块（NQM）进行缓存。

网络队列管理模块在接口空闲时会将队列中的 TSntag 和 bufid 读出并输出给网络发送模块。

网络发送模块（NTX）接收到 TSntag 和 bufid 后，将 bufid 转化为报文从集中缓存区中读取的基地址，根据此地址从报文集中缓存模块中读出报文，在接收到最后一拍数据时，将 bufid 传给报文集中缓存模块进行释放；把报文的位宽由 134bit 转换为 8bit 并进行时钟域切换（从架构内部处理时钟域到 GMII 发送时钟域）后从 GMII 接口输出报文。针对 PTP 报文，还需要计算 PTP 报文在 TSE 中传输的透明时钟并累加到 PTP 报文的透明时钟域中。

2.2.2 网络口进主机口出的帧处理流程

网络口输入的报文先在网络接收模块（NRX）进行跨时钟域处理，并在每拍数据中增加 1bit 的头尾标识位，然后根据端处理逻辑所处的阶段来决定是否传输报文：若端处理逻辑处于初始化阶段，丢弃所有报文；若端处理逻辑处于配置阶段，只传输配置报文；端处理逻辑处于时钟同步阶段，传输非时间敏感报文；端处理逻辑处于正常工作阶段，接收所有报文。

帧解析模块（FPA）将接收到的报文写到一个 134bit 寄存器中，把报文位宽由 9bit 转换为 134bit，当寄存器写满时便将数据输出给输入缓存接口模块；将报文的 TSNtag 以及报文集中缓存模块分配的 bufid 组成报文描述符，并对 RC 报文和 BE 报文进行监管，若 bufid 剩余数量少于 RC 报文阈值时，将 RC 报文描述符和 BE 报文描述符均丢弃，若 bufid 剩余数量少于 BE 报文阈值时，将 BE 报文描述符丢弃；然后根据报文以太网类型对报文描述符进行分派，将 ARP 报文描述符或映射后的 IP 报文描述符传输给主机接口。

输入缓存接口模块（IBI）将 bufid 转换为报文在集中缓存区中缓存基地址，将报文写到集中缓存区中，每接收到一拍 134bit 数据时，便向报文集中缓存模块输出写请求，在收到报文集中缓存模块的响应后，完成一拍数据的写入。

主机队列管理模块（HQM）将接收到的 TSNtag 和 bufid 写到队列中进行缓存，在接收到帧逆映射模块（FIM）传来的准备就绪信号后，将 bufid 和 TSNtag 中的 flowID 输出给帧逆映射模块。

帧逆映射模块用 flowid 去查找逆映射表，将 bufid、查表是否命中、以及查表命中时的查表结果 DMAC 输出给主机发送模块（HTX）。

主机发送模块检查在帧逆映射模块中 flowID 查找逆映射表是否命中，若查表未命中，则将 bufid 释放给报文集中缓存模块，无需从报文集中缓存模块中读出报文；若查表命中，则将接收到的 bufid 转化为报文读取的基地址，从报文集中缓存模块（PCB）的缓存区中读出报文，在接收到最后一拍数据时，将 bufid 传给报文集中缓存模块

进行释放；在读出报文第一拍数据时，将查逆映射表的结果 DMAC 替换报文 TSntag，将以太网类型由 0x1800 改为 0x0800；然后把报文位宽由 134bit 转换为 8bit，将报文传输时钟域由架构内部处理时钟域转换为 GMII 发送时钟域，接下来从 GMII 接口输出报文。

2.2.3 网络口进网络口出的帧处理流程

网络口输入的报文先在网络接收模块(NRX)进行跨时钟域处理，并在每拍数据中增加 1bit 的头尾标识位；针对 PTP 报文，需要记录接口接收 PTP 报文的时间戳，并存放在 TSntag 中；然后根据端处理逻辑所处的阶段来决定是否传输报文：若端处理逻辑处于初始化阶段，丢弃所有报文；若端处理逻辑处于配置阶段，只传输配置报文；端处理逻辑处于时钟同步阶段，传输非时间敏感报文；端处理逻辑处于正常工作阶段，接收所有报文。

帧解析模块(FPA)将接收到的报文写到一个 134bit 寄存器中，把报文位宽由 9bit 转换为 134bit，当寄存器写满时便将数据输出给输入缓存接口模块；将报文的 TSntag 以及报文集中缓存模块分配的 bufid 组成报文描述符，并对 RC 报文和 BE 报文进行监管，若 bufid 剩余数量少于 RC 报文阈值时，将 RC 报文描述符和 BE 报文描述符均丢弃，若 bufid 剩余数量少于 BE 报文阈值时，将 BE 报文描述符丢弃；然后根据报文以太网类型对报文描述符进行分派，将来自 HCP 的映射后 PTP 报文描述符或 TSMP 报文描述符传输给网络接口（连接交换机），将来自交换机的映射后 PTP 报文描述符或 TSMP 报文描

述符传输给网络接口（连接 HCP）。

输入缓存接口模块（IBI）将 bufid 转换为报文在集中缓存区中缓存基地址，将报文写到集中缓存区中，每接收到一拍 134bit 数据时，便向报文集中缓存模块输出写请求，在收到报文集中缓存模块的响应后，完成一拍数据的写入。

网络队列管理模块（NQM）在接口空闲时会将队列中的 TSNTag 和 bufid 读出并输出给网络发送模块。

网络发送模块（NTX）接收到 TSNTag 和 bufid 后，将 bufid 转化为报文从集中缓存区中读取的基地址，根据此地址从报文集中缓存模块中读出报文，在接收到最后一拍数据时，将 bufid 传给报文集中缓存模块进行释放；把报文的位宽由 134bit 转换为 8bit 并进行时钟域切换（从架构内部处理时钟域到 GMII 发送时钟域）后从 GMII 接口输出报文。针对 PTP 报文，还需要计算 PTP 报文在 TSE 中传输的透明时钟并累加到 PTP 报文的透明时钟域中。

附录 A：数据格式定义

● TSNTag 格式

在流量发送端的网卡内部需要根据报文七元组（目的 mac、type、IP 五元组）对时间敏感、带宽预约、尽力转发流量进行分类映射。将分类映射的结果与原报文的 DMAC 字段进行替换，以此进行 TSN 网络的交换，直到接收端的网卡内部进行 DMAC 还原。被替换的 DMAC 字段被定义成 TSNTag。

表 A-1 分类映射关键字 Key

| 位宽 | 名称 | 描述 |
|----|----------|----------|
| 48 | DMAC | 报文目的 MAC |
| 16 | ETHTYPE | 报文以太网类型 |
| 8 | protocol | 报文协议类型 |
| 32 | Sip | 报文源 ip |
| 32 | Dip | 报文目的 ip |
| 16 | Sport | 报文源端口 |
| 16 | Dport | 报文目的端口 |

因同步报文的 TSNTag 中“seq_id”、“frag_id”、“inject_addr”、“submit_addr”信息是无用的，因此可以将时间同步报文的这些字段用来存放架构的接收时间戳信息。而其他非时间同步报文的架构接收时间戳信息是无用的，因此可以延用这些字段的信息。

表 A-2 时间同步报文的 TSNTag

| 位宽 | 名称 | 位置 | 描述 |
|----|---------------|---------|---|
| 3 | Flow type | [47:45] | 流类型。 100: 同步报文（其他报文的格式如表 A-3） |
| 14 | Flow id/IMAC | [44:31] | 静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换架构命中同一条表项。 |
| 12 | Reserve | [30:19] | 保留 |
| 19 | Rx_timestamps | [18:0] | 架构接收到时间同步报文的本地时间信息，用于架构发送报文时计算透明时钟。 |

表 A-3 非时间同步报文的 TSNTag

| 位宽 | 名称 | 位置 | 描述 |
|----|--------------|---------|---|
| 3 | Flow type | [47:45] | 流类型。 000:ST 分组 001:ST 分组 010: ST 分组 011: RC 分组 101: NMAC 分组 110: BE 分组 111: 需重组的分组 |
| 14 | Flow id/IMAC | [44:31] | 静态流量使用 flowID，每条静态流分配一个唯一 flowID，动态流使用 imac 地址，imac 地址相同的则在交换架构命中同一条表项。 |

| | | | |
|----|-------------|---------|------------------------------|
| 16 | Seq id | [30:15] | 用于标识每条流中报文的序列号 |
| 1 | Frag flag | [14] | 用于标识分片后的尾。0：分片后的中间报文 1：尾拍 |
| 4 | Frag ID | [13:10] | 用于表示当前分片报文在原报文中的分片序号 |
| 5 | inject addr | [9:5] | ST 流在源端等待发送调度时缓存地址 |
| 5 | submit addr | [4:0] | ST 流在终端等待接收调度时缓存地址 |

● 内部传输的 pkt 数据格式

在模块之间传输的有两种格式，一种位宽为 9bit；另一种位宽为 134bit。具体区别如下：

pkt_data 位宽为 9 位，包含 1bit 头尾标志、8bit 报文数据。头尾标志：1 表示报文头/尾数据；0 标识报文体中间数据。具体如图 A-1 所示：

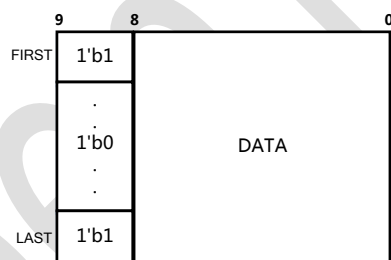


图 A-1 pkt_data 数据格式 1

pkt_data 位宽为 134 位，包含 2bit 头尾标志、4bit 无效字节、128bit 报文数据。头尾标志：01 表示报文头；11 表示报文体中间数据；10 表示报文尾。无效字节用于标识报文尾中无效的字节数。具体如图 A-2 所示：

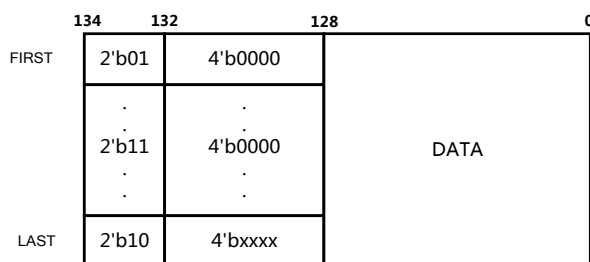


图 A-2 pkt_data 数据格式 2

附录 B：内部表项格式定义

● 转发表

转发表的格式如表 B-1 所示。

表 B-1 转发表数据格式

| 名称 | 含义 | 备注 |
|--------------|------------------------------------|---------------------|
| outport[8:0] | 输出端口号(bitmap)，总共 8 个网络接口与 1 个主机接口。 | 使用 RAM 进行实现，深度为 16K |

转发表设计在转发查表(FLT)模块，该模块会根据接收到的描述符提取出 flow_id，再将 flow_id 作为查表地址进行查找转发表。查表结果为输出端口号，其中某位为高则代表往某个对应的输出端口进行转发，本模块根据查表结果将描述符内容转发给对应的输出端口。转发表的深度为 16K，代表由 TSNSwifth 组成的整个网络最多能够支持 16K 条流的转发。

● 门控表

门控表的格式如表 B-2 所示：

表 B-2 门控表数据格式

| 名称 | 含义 | 备注 |
|-----------------------|------------------|----------------------|
| gate_ctrl_vector[7:0] | 门控向量，对应 8 个门控信息。 | 使用 RAM 进行实现，深度为 1024 |

门控表设计在交换发送处理逻辑中的队列门控(QGC)模块，该模块根据接收到的时间槽信息以及时间槽切换信号来进行查表。本模块每接收到一个时间槽切换信号，便将时间槽的 ID 作为查表地址进行查找门控表，查表结果为门控向量，对应 8 个门控信号，当某位为高

则代表打开对应的某个队列的门控，网络输出调度(NOS)模块根据该门控进行对队列进行调度。门控表的深度为 1024，代表 TSNSwitch 最多能够支持 1024 个时间槽做为一个周期。

附录 C：内部寄存器定义

架构内部可配置地址空间主要有两部分，包括：MDID 模块号和真实地址空间，其中 MDID 模块号主要用来区分不同模块，而后 20 位为各个模块使用的地址空间。地址的第 19bit 位用于区别地址类型，控制/表项寄存器可读可写，调试和版本寄存器只读，每个模块的地址空间为 1024k,其中可读可写和只读寄存器各有 512k。具体地址含义如下。

表 C-1 地址格式

| ADDR[26:0] | | |
|--------------|----------|--------------------|
| MDID[26:20] | ADDR[19] | ADDR[18:0] |
| MDID : 0-127 | 0 | 该模块的控制寄存器,表项等,可读可写 |
| | 1 | 只读 |

每个处理模块的 MDID 号分配如下：

表 C-2 模块中的 MDID 和地址

| 处理模块 | CSM | TIS | TSS | QGC | GTS | FLT |
|------|-------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| MDID | 0x0 | 0x1 | 0x2 | 0x3-0xa | 0xb | 0xc |
| 地址 | 0x0-0xfffff | 0x100000-0x1fffff | 0x200000-0x2fffff | 0x300000-0xafffff | 0xb00000-0xbfffff | 0xc00000-0xcfffff |

● CSM 模块

地址范围为 Addr 0x0-0xfffff。

表 C-3 CSM 模块寄存器

| Addr | Data | | | |
|--------------------|------------------------|---------|--------|-------|
| | [31:24] | [23:16] | [15:8] | [7:0] |
| 0x0 | offset_l | | | |
| 0x1 | offset_h | | | |
| 0x2 | time_slot | | | |
| 0x3 | cfg_finish | | | |
| 0x4 | port_type | | | |
| 0x5 | qbv_or_ach | | | |
| 0x6 | report_type | | | |
| 0x7 | report_en | | | |
| 0x8 | inject_slot_period | | | |
| 0x9 | submit_slot_period | | | |
| 0xa | report_period | | | |
| 0xb | offset_period | | | |
| 0xc | rc_regulation_value | | | |
| 0xd | be_regulation_value | | | |
| 0xe | unmap_regulation_value | | | |
| 0xf ~ 0xffff | reserve | | | |

表 C-4 寄存器的具体含义

| name | bit | R/W | description | default |
|------------|-------|-----|-------------------------------|---------|
| offset_l | 31:17 | R/W | 代表时间偏移的高位值的低 15 位，表示毫秒 | 0 |
| | 16:0 | R/W | 时间偏移的低位，表示拍数 | 0 |
| offset_h | 31:17 | R/W | 保留位 | 0 |
| | 16 | R/W | 代表时间偏移的正负值，1 代表正值，如果为 0，则代表负值 | 0 |
| | 15:0 | R/W | 代表时间偏移的高位值的高 16 位，表示毫秒 | 0 |
| time_slot | 31:11 | R/W | 保留 | |
| | 10:0 | R/W | 时间槽大小 | 0 |
| cfg_finish | 31:2 | R/W | 保留 | 0 |
| | 1:0 | R/W | 配置完成寄存器，0 代表架构正在初始化，不接 | 0 |

| name | bit | R/W | description | default |
|--------------------|-------|-----|---|---------|
| | | | 收任何报文， 1 代表初始化完成，可以接收 NMAC 配置报文 2 代表配置完成，可以接收除 ST 报文的任何报文 3 代表可以接收任何报文 | |
| port_type | 31:8 | R/W | 保留 | 0 |
| | 7:0 | R/W | 网络端口类型寄存器，架构共有 8 个网络端口，寄存器的 0-7 位分别代表 0-7 端口的类型，1 代表非合作类型，处理标准以太网类型的报文，0 代表合作类型，处理 TSN 报文 | 0 |
| qbv_or_ach | 31:1 | R/W | 保留 | 0 |
| | 0 | R/W | 调度模式选择信号，网络输出逻辑中的调度机制是 QBV 模式还是 QCH 模式 0 代表 QBV 模式；1 代表 QCH 模式 | 0 |
| report_type | 31:16 | R/W | 保留 | 0 |
| | 15:0 | R/W | 上报类型，具体参考附录 D | 0 |
| report_en | 31:1 | R/W | 保留 | 0 |
| | 0 | R/W | 上报使能信号，配置与状态管理模块是否进行周期性上报 0 代表不上报；1 代表上报 | 0 |
| inject_slot_period | 31:12 | R/W | 保留 | 0 |
| | 10:0 | R/W | 注入时间槽周期，架构内部时间槽切换的周期值 配置的值范围：1-1024 个 | 0 |
| submit_slot_period | 31:12 | R/W | 保留 | 0 |
| | 10:0 | R/W | 提交时间槽周期，架构内部时间槽切换的周期值 配置的值范围：1-1024 个 | 0 |
| report_period | 31:12 | R/W | 保留 | 0 |
| | 11:0 | R/W | 上报周期，配置与状态管理模块上报的周期值 配置的值范围：1 (ms) 或 1000 (ms) | 0 |
| offset_period | 31:24 | R/W | 保留 | 0 |
| | 23:0 | R/W | offset 补偿的配置周期 | |

| name | bit | R/W | description | default |
|------------------------|------|-----|---|---------|
| rc_regulation_value | 31:9 | R/W | 保留 | 0 |
| | 8:0 | R/W | RC 流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃 RC 报文 | |
| be_regulation_value | 31:9 | R/W | 保留 | 0 |
| | 8:0 | R/W | BE 流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃 BE 报文和 RC 报文 | |
| unmap_regulation_value | 31:9 | R/W | 保留 | 0 |
| | 8:0 | R/W | 非映射流的监管阈值，当 BUFID 的剩余个数小于该值，开始丢弃非映射报文 | |
| reserve | 31:9 | R/W | 保留 | 0 |

● TIS 模块

地址范围为 Addr 0x100000-0x1fffff。

表 C-5 地址格式

| Addr | Data | | | |
|-----------------------------------|--|---------|--------|-------|
| | [31:24] | [23:16] | [15:8] | [7:0] |
| send_table_N 0x100000-0x1003ff | ST 报文发送时刻表每项内容，N=0、1、...、1023 send_table_0 表示第 0 个发送表 | | | |
| 0x100400-0x1fffff | 保留 | | | |

表 C-6 寄存器的具体含义

| name | bit | R/W | description | default |
|-----------------|------|-----|----------------------|---------|
| send_table_0 | 16 | R/W | 保留 | 0 |
| | 15 | R/W | 表项有效位，0 代表无效，1 代表有效 | 0 |
| | 14:5 | R/W | ST 流在一个应用周期内的注入时间槽 | 0 |
| | 4:0 | R/W | TSNTag 中的“send addr” | 0 |
| | | | | |
| send_table_1023 | 16 | R/W | 保留 | 0 |

| | | | | |
|------------------|------|-----|----------------------|---|
| | 15 | R/W | 表项有效位，0 代表无效，1 代表有效 | 0 |
| | 14:5 | R/W | ST 流在一个应用周期内的注入时间槽 | 0 |
| | 4:0 | R/W | TSNTag 中的“send addr” | 0 |
| 0x100400-0x1ffff | | | 保留 | |

● TSS 模块

地址范围为 Addr 0x200000-0x2ffff。

表 C-7 地址格式

| Addr | Data | | | |
|-------------------------------------|--|---------|--------|-------|
| | [31:24] | [23:16] | [15:8] | [7:0] |
| submit_table_N 0x200000-0x2003ff | ST 报文提交时刻表每项内容，N=0、1、...、1023 submit_table_0 表示第 0 个提交表 | | | |
| 0x200400-0x2ffff | 保留 | | | |

表 C-8 寄存器的具体含义

| name | bit | R/W | description | default |
|-------------------|-------|-----|----------------------|---------|
| submit_table_0 | 31:16 | R/W | 保留 | 0 |
| | 15 | R/W | 表项有效位，0 代表无效，1 代表有效 | 0 |
| | 14:5 | R/W | ST 流的提交时间槽 | 0 |
| | 4:0 | R/W | TSNTag 中的“send addr” | 0 |
| | | | | |
| submit_table_1023 | 31:16 | R/W | 保留 | 0 |
| | 15 | R/W | 表项有效位，0 代表无效，1 代表有效 | 0 |
| | 14:5 | R/W | 当前 Slot | 0 |
| | 4:0 | R/W | TSNTag 中的“send addr” | 0 |

● QGC 模块

地址范围为 Addr 0x300000-0xafffff，其中 0x300000-0x3fffff 表示第一个端口的门控表，以此类推，共有 8 个端口门控。

表 C-9 地址格式

| Addr | Data | | | |
|---|--|---------|--------|-------|
| | [31:24] | [23:16] | [15:8] | [7:0] |
| port0_gate_table_N 0x300000-0x3003ff | 0 号端口的门控表，N=0、1、...、1023，输出门控 port0_gate_table_0 表示 0 号端口的第一个时刻的门控状态 | | | |
| port1_gate_table_N 0x400000-0x4003ff | 1 号端口的门控表，N=0、1、...、1023 | | | |
| port2_gate_table_N 0x500000-0x5003ff | 2 号端口的门控表，N=0、1、...、1023 | | | |
| port3_gate_table_N 0x600000-0x6003ff | 3 号端口的门控表，N=0、1、...、1023 | | | |
| port4_gate_table_N 0x700000-0x7003ff | 4 号端口的门控表，N=0、1、...、1023 | | | |
| port5_gate_table_N 0x800000-0x8003ff | 5 号端口的门控表，N=0、1、...、1023 | | | |
| port6_gate_table_N 0x900000-0x9003ff | 6 号端口的门控表，N=0、1、...、1023 | | | |
| port7_gate_table_N 0xa00000-0xa003ff | 7 号端口的门控表，N=0、1、...、1023 | | | |

表 C-10 寄存器的具体含义

| name | bit | R/W | description | default |
|-----------------------|------|-----|--|---------|
| port0_gate_table_0 | 31:8 | R/W | 保留 | 0 |
| | 7:0 | R/W | 0-7 位分别代表 0-7 共 8 个队列的门控状态，0 代表该队列的门控关闭，1 代表开启 | 0 |
| | | | | |
| port7_gate_table_1023 | 31:8 | R/W | 保留 | 0 |
| | 7:0 | R/W | 0-7 位分别代表 0-7 共 8 个队列的门控状态，0 代表该队列的门控关闭，1 代表开启 | 0 |

● FLT 模块

地址范围为 Addr 0xc00000-0xcfffff。

表 C-11 地址格式

| Addr | Data |
|------|------|
|------|------|

| | [31:24] | [23:16] | [15:8] | [7:0] |
|-------------------|--|---------|--------|-------|
| 0xc00000-0xc03fff | forward_table_N，表示转发表，N=0,1,2, ...16383，forward_table_0 表示第 0 个转发表 | | | |
| 0xc04000-0xcfffff | 保留 | | | |

表 C-12 寄存器的具体含义

| name | bit | R/W | description | default |
|---------------------|-------|-----|--|---------|
| forward_table_0 | 31:16 | R/W | 保留 | 0 |
| | 8:0 | R/W | 转发表的内容，使用 bitmap 的形式，0-8 位分别代表向 0-8 号端口，每位的值 0 代表不向该端口转发，1 代表向该端口转发 | 0 |
| | | | | |
| forward_table_16383 | 31:16 | R/W | 保留 | 0 |
| | 8:0 | R/W | 转发表的内容，使用 bitmap 的形式，0-8 位分别代表向 0-8 号端口，每位的值为 0 代表不向该端口转发，1 代表向该端口转发 | 0 |
| 0xc04000-0xcfffff | | | 保留 | |

附录 D：NMAC 报文格式

● 配置报文格式

配置报文格式：在报文中用 count 字段（8bit）表示报文中包含的配置条目数，报文最小为 64 字节，最后不够 64 字节的报文需要补零。NMAC 命令在以太网报文中的封装如图 D-1 所示。

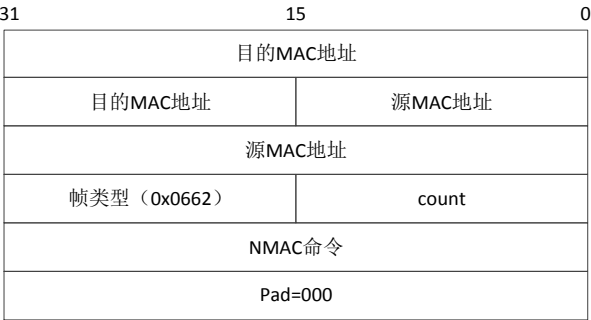


图 D-1 NMAC 配置报文格式

NMAC 命令的格式如图 D-2 所示。

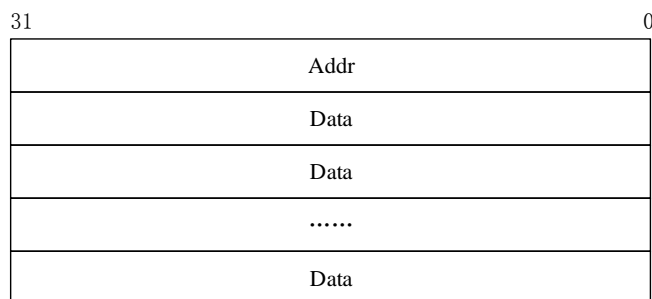


图 D-2 NMAC 命令格式

当配置的寄存器数量为 1，NMAC 命令就包括 32bit 的 ADDR 和 32bit 的 DATA；当配置的寄存器数量为 N (N>1)，NMAC 命令就包括 32bit ADDR 和 N*32bit 的 DATA，第一个 DATA 以 ADDR 作为 RAM 写地址，第二个以及后续 DATA 以 ADDR 循环加 1 作为 RAM 写地址。

● 上报报文格式

上报报文格式：其中报文类型为 NMAC 报文（0x1662）。

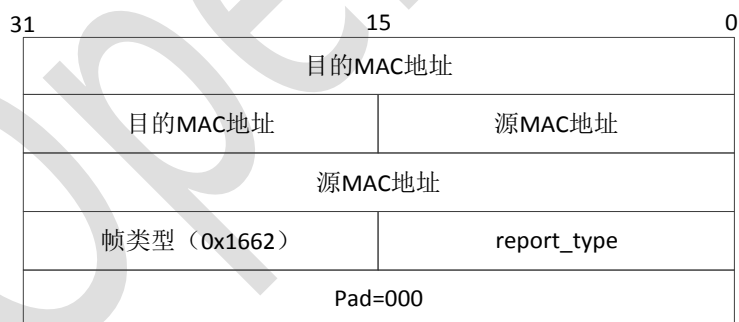


图 D-3 NMAC 上报报文格式

表 D-1 上报类型格式

| 上报类型（16bit） | | 含义 |
|-----------------|---------|---|
| 高 6bit | 低 10bit | |
| 000000 单个寄存器 | 0 | 配置的单个寄存器，包含配置完成寄存器、端口状态寄存器、时间槽大小寄存器、时间偏移寄存器、上报周期寄存器、上报类型寄存器、应用周期寄存器 |

| 上报类型（16bit） | | 含义 |
|--------------------------------------|---------|-------------------|
| 高 6bit | 低 10bit | |
| 000001 转发表 | 0 | 第 0-63 条转发表 |
| | 1 | 第 64-127 条转发表 |
| | 2-255 | 第 128-16383 条转发表 |
| 000010 注入时刻表 | 0 | 第 0-63 条注入时刻表 |
| | 1 | 第 64-127 条注入时刻表 |
| | 2-15 | 第 128-1023 条注入时刻表 |
| 000011 提交时刻表 | 0 | 第 0-63 条注入时刻表 |
| | 1 | 第 64-127 条注入时刻表 |
| | 2-15 | 第 128-1023 条注入时刻表 |
| 000100-001 011 P0-P7 输出 门控表 | 0 | 第 0-63 条注入时刻表 |
| | 1 | 第 64-127 条注入时刻表 |
| | 2-15 | 第 128-1023 条注入时刻表 |
| 001100 xx_state | 0 | 暂未开发 |

单个寄存器，report_type 高 6bit 为 0000，低 10bit 为 0。

转发表上报报文格式，report_type 高 6bit 为 000001，低 10bit 为上报的第几块，转发表一共有 16K 条，每条转发表占用 2 字节(9bit)，因此每个报文可以携带 64 条，总共需要 256 个报文。

注入时刻表上报报文格式，report_type 高 6bit 为 000010，低 10bit 为上报的第几块，注入时刻表一共有 1024 条，每条转发表占用 2 字节（9bit），因此每个报文可以携带 64 条，总共需要 16 个报文。

提交时刻表上报报文格式，report_type 高 6bit 为 000011，低 10bit 为上报的第几块，注入时刻表一共有 1024 条，每条转发表占用 2 字节（9bit），因此每个报文可以携带 64 条，总共需要 16 个报文。

门控表按照端口划分，每个门控占用一块 RAM，共有 8 个端口。每个端口两块 RAM，总共需要 16 块 RAM。

附录 E: command/command_ack 命令格式

表 E-1 command/command_ack 命令格式

| 位置 | 位宽 | 名称 | 说明 |
|-----------|-----|----------------|---|
| [203:180] | 8 | node_id | 该字段用来标识对哪个节点进行读写。每个 TSE 或 TSS 都有一个唯一的节点 ID。该字段在 TSN 网卡+TSN 交换机模式下使用到。 |
| [179:172] | 8 | dest_module_id | 该字段用来标识对一个节点内的哪个模块进行控制。TSE 或 TSS 内部每个子模块都有一个唯一的模块 ID |
| [171:168] | 4 | type | 4'b0001:寄存器或表项的写命令; 4'b0010:寄存器或表项的读命令; 4'b0110:寄存器或表项的读响应。 |
| [167:152] | 16 | addr | 寄存器或表项的读/写地址 |
| [151:0] | 152 | data | 寄存器或表项的读/写数据; 其中五元组映射表的表项位宽最大, 为 152bit |